PRESENTATION OUTLINE: Exploring how OpenMP can improve application performance

Dehui Yu
School of Computer Science
University of Ottawa
Ottawa, Canada M1W 0A5
dyu105@uottawa.ca

November 30, 2022

1 Introduction (Slide 3-9)

- Introduction to Parallel Programming[4]
- Introduction to OpenMP API[11]
- Introduction to Monto Carlo method[5][7]

2 Literature Review (Slide 10-14)

- Why OpenMP becomes the standard API for Parallel Programming?[4][12][3]
- Examples using OpenMP for performance improvement[10][8]

3 Problem Statement (Slide 15)

- By simply adding more threads, will the performance always improve?[1]
- How many threads are best for parallel applications?

4 Case Study 1 (Slide 16-22)

- Experimental Environment
- Calculating PI using OpenMP based on Monte Carlo algorithm[9][6]
- Source code
- Performance Analysis

5 Case study 2 (Slide 23-27)

- Experimental Environment
- Matrix multiplication using OpenMP[2]
- Blocked matrix multiplication using OpenMP
- Performance Analysis

6 Summary (Slide 28)

- Summary about my presentation

7 Discussion (Slide 29)

- Discussion for students

References

[1] Eduard Ayguadé, Nawal Copty, Alejandro Duran, Jay Hoeflinger, Yuan Lin, Federico Massaioli, Xavier Teruel, Priya Unnikrishnan, and Guansong Zhang. The design of openmp tasks. *IEEE Transactions on Parallel and Distributed Systems*, 20(3):404–418, 2008.

[2] K Mani Chandy. Parallel program design. In *Opportunities and Constraints of Parallel Computing*, pages 21–24. Springer, 1989.

[3] Barbara Chapman, Gabriele Jost, and Ruud Van Der Pas. *Using OpenMP: portable shared memory parallel programming*. MIT press, 2007.

[4] L. Dagum and R. Menon. Openmp: an industry standard api for shared-memory programming. *IEEE Computational Science and Engineering*, 5(1):46–55, 1998.

[5] Thiago de Jesus Oliveira Duraes, Paulo Sergio Lopes de Souza, Guilherme Martins, Davi Jose Conte, Naylor Garcia Bachiega, and Sarita Mazzini Bruschi. Research on parallel computing teaching: state of the art and future directions. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, 2020.

[6] Robert L Harrison. Introduction to monte carlo simulation. In *AIP conference proceedings*, volume 1204, pages 17–21. American Institute of Physics, 2010.

[7] Frederick James. Monte carlo theory and practice. *Reports on progress in Physics*, 43(9):1145, 1980.

[8] Honghoon Jang, Anjin Park, and Keechul Jung. Neural network implementation using cuda and openmp. In *2008 Digital Image Computing: Techniques and Applications*, pages 155–161, 2008.

[9] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[10] Jeffrey Neal, Timothy Fewtrell, and Mark Trigg. Parallelisation of storage cell flood models using openmp. *Environmental Modelling Software*, 24(7):872–877, 2009.

[11] Thomas Rauber and Gudula Rünger. *Parallel programming*. Springer, 2013.

[12] Alejandro Rico Carro, Isaac Sánchez Barrera, Jose Joao, Joshua Randall, Marc Casas, and Miquel Moretó. *On the Benefits of Tasking with OpenMP*, pages 217–230. 08 2019.